Introduction

This getting started guide for the OSD3358-SM-RED board is intended to serve as a quick reference to getting the OSD3358-SM RED board up and running.

This document will be updated as required to improve or add information. Please make sure to look for updates and sign up for document change notifications on Octavo Systems website to get up to date info.

Revision History

Revision	Details	Date	Author
Initial Revision		9-7-2017	Neeraj Dantu

OSD3358-SM-RED Getting Started Guide

Rev. 1.0 9/7/2017

Contents

1 Features

The following figure highlights the features of the OSD3358-SM-RED platform (RED Board) and where they are located on the board.

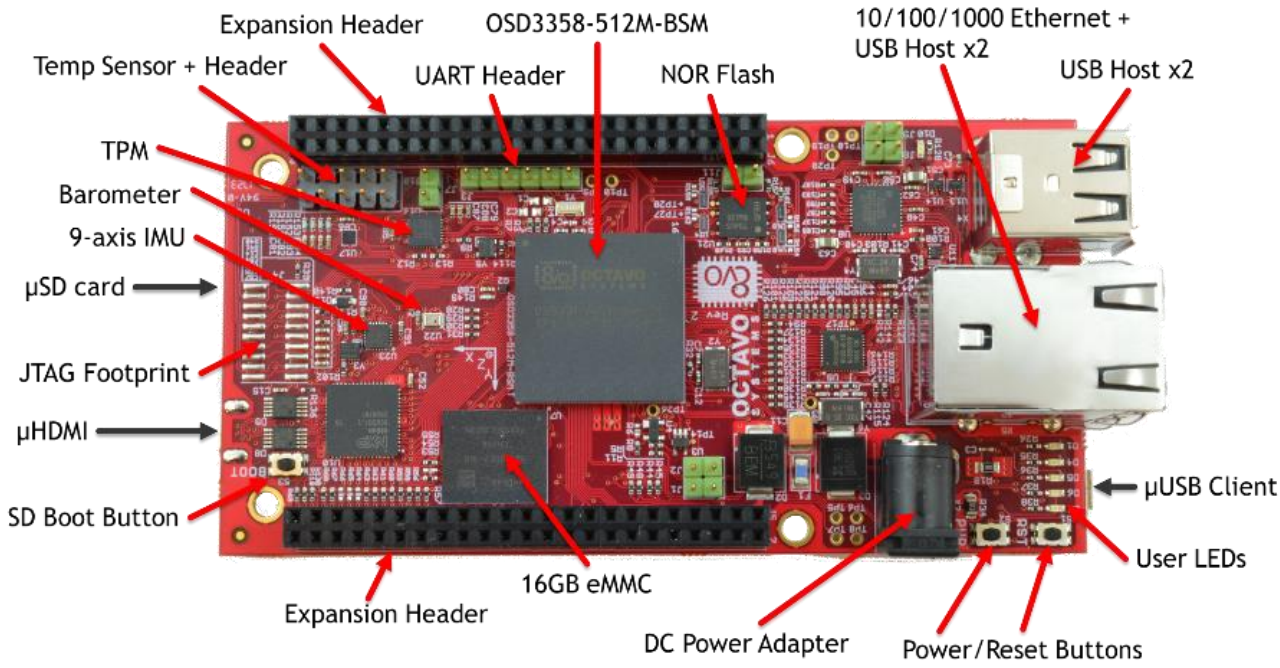

Figure 1: OSD3358-SM-RED board features

2 Unboxing your board

The contents of the OSD3358-SM-RED platform are listed below:

1. OSD3358-SM-RED

Figure 2: OSD3358-SM-RED

2. Micro-USB to USB cable

Figure 3: USB cable

3. Getting started card

Figure 4: OSD3358-SM-RED platform getting started card

3 Powering the OSD3358-SM-RED

Figure 5 shows the OSD3358-SM-RED and its power inputs. The board can be powered through any of three inputs:

a. Micro-USB cable connected to the micro-USB port
b. 5V AC adapter connected to the 2.5 mm x 5.5 mm barrel jack
c. Single cell Li-Ion or Li-Polymer battery connected to the battery header

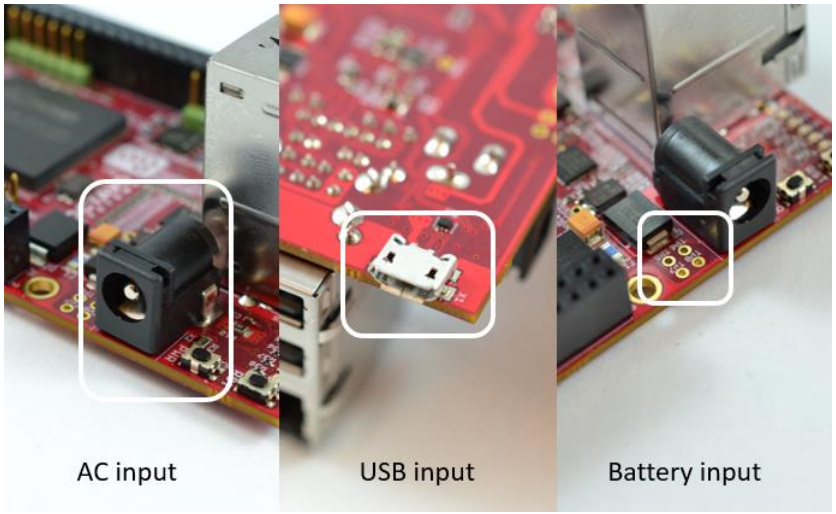

Figure 5: Picture of RED board highlighting power inputs

3.1 Power up procedure

The following steps describe the power up sequence of the board when powered via USB, AC Adapter or battery inputs.

1. Connect your preferred power source:
    a. USB:
        i. Connect the micro-USB side of the USB cable to the micro-USB port X1
        ii. Connect the USB side to one of the USB ports on a laptop/computer or other USB power source.

Figure 6: Powering the board using a USB cable

b. AC adapter:
    i. Plug in the 5V AC adapter with 2.5 mm x 5.5 mm barrel into the power connector X2.

Figure 7: Powering the RED board using an AC-DC adapter

c. Battery input terminals:
    i. Connect the battery input terminals to the battery terminals as shown below.

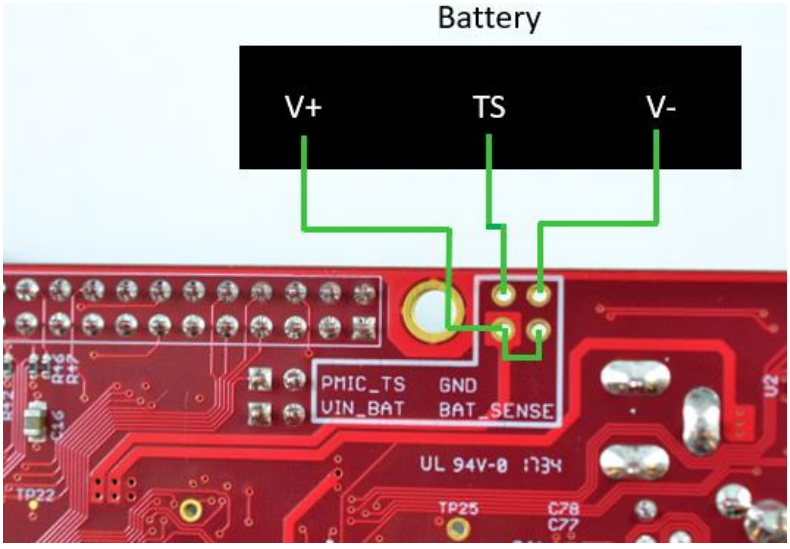

Figure 8: OSD3358-SM-RED battery connections (back-side of board)

NOTE: Notes on connecting the PMIC_TS input of the SiP can be found in "Battery pack temperature monitoring" section of the TPS65217C PMIC datasheet. If the battery does not have a temperature output, the battery temperature monitoring mechanism of the PMIC can be bypassed by connecting a 10KOhm resister between PMIC_TS (TP7) of the board and GND.

2. LED D1 will turn on immediately after power is applied to the board.

Figure 9: RED board boot up - LED D1 ON

3. LEDs D4 – D7 turn on in the sequence indicated below indicating Linux is booting.

Figure 10: RED board boot up - LEDs D4 - D7 turn ON

4 OSD3358-SM-RED Use Cases

The OSD3358-SM-RED Platform can be used in 3 different modes. A description of each mode and instructions on how to set up each mode is described below.

4.1 Standalone boot up

The board can be used as a standalone platform for software development. To boot it up as a standalone system, the following components will be needed.

1. 5V AC adapter
2. HDMI/DVI-D monitor
3. Micro HDMI to HDMI cable and/or HDMI to DVI-D adapter
4. Wired/wireless USB keyboard and mouse
5. Ethernet cable or WiFi USB adapter connected to a network for internet access

The following steps describe the procedure to setup the board as a standalone software development tool.

1. Connect the micro HDMI connector of the HDMI cable to X6 connector on board.
2. Connect the HDMI connector of the HDMI cable into the monitor. HDMI to DVI-D convertor may be necessary if the monitor only has a DVI-D input.

Figure 11: Inserting HDMI cable into the monitor

3. Connect the wired/wireless USB keyboard and mouse inputs to any of the four USB ports available on X4 and X5 connectors of the board.

Figure 12: Wireless adapter for keyboard and mouse plugged in to RED board

4. If internet connectivity is required:
    a. a WiFi-USB adapter can be plugged into one of the available USB ports on X4 or X5 connector. A list of compatible adapters can be found at http://www.elinux.org/Beagleboard:BeagleBoneBlack#WIFI_Adapters. Procedures to connect to a WiFi network using the adapter can be found online.
    OR
    b. An ethernet cable can be plugged into the ethernet connector receptacle on X4.

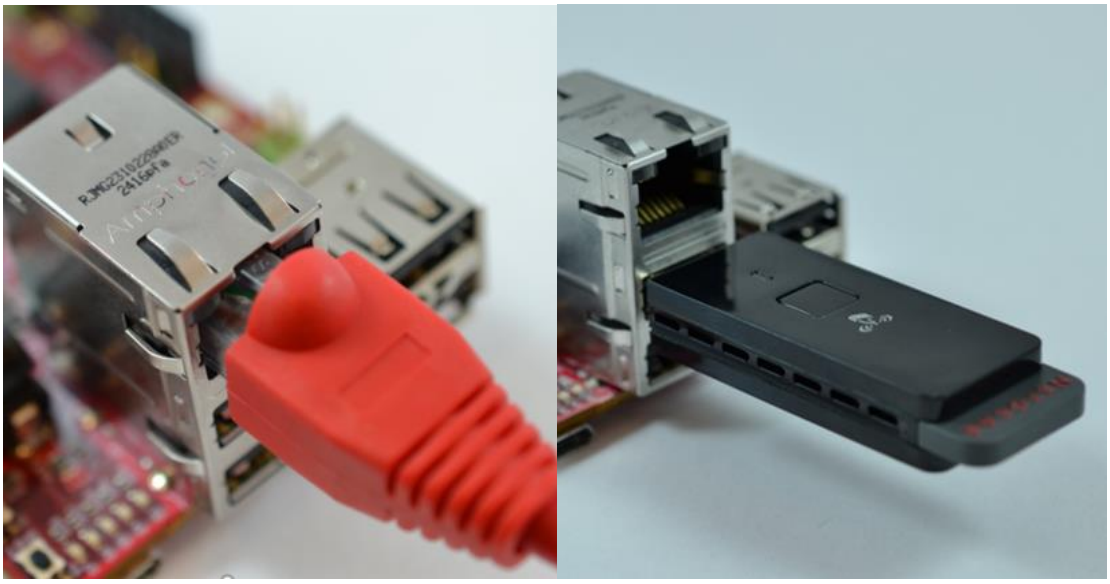

Figure 13: Ethernet cable/Wi-Fi dongle plugged in to the RED board

5. Plug the 5V AC adapter into the power connector X2.

Figure 14: AC-DC adapter plugged in to RED board

6. The board will go through the default boot up process
7. After the board boots up, the monitor screen should show a Linux desktop environment as shown below.

Figure 15: OSD3358-SM-RED desktop environment

8. Click on the Start>System Tools>Q Terminal to open a command terminal.

Figure 16: Shell command terminal on desktop environment

4.2 Booting as USB client connected to a computer/laptop

The OSD3358-SM-RED platform can also be powered using a computer/laptop's USB port. The micro-USB to USB cable provided in the box can be used to connect the RED board to a computer/laptop. Unlike the standalone mode, this setup does not require additional hardware. The following steps describe the procedure to power the board as a USB slave and set up the development environment.

1. Connect the micro-USB side of the USB cable to the micro-USB port X1
2. Connect the USB side to one of the USB ports on a laptop/computer.

Figure 17: Powering RED board using a USB cable

3. Wait for the board to show up as a mass storage device on the computer/laptop

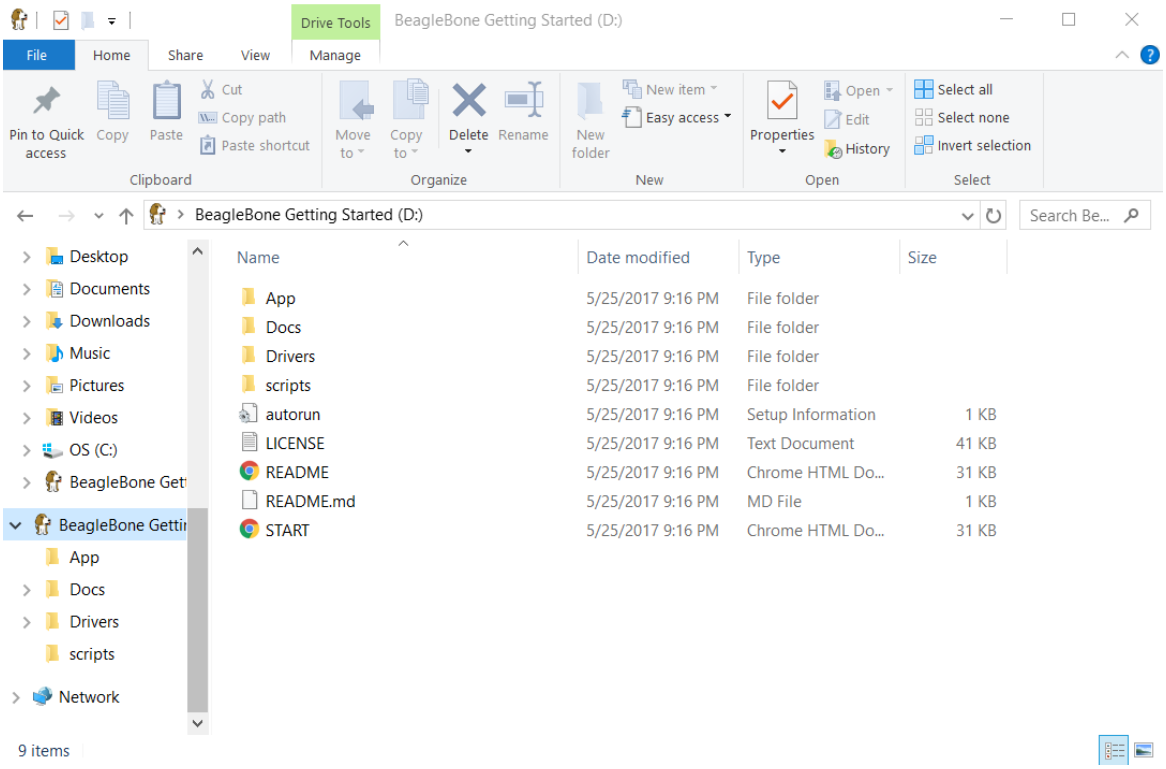

Figure 18: OSD3358-SM-RED USB mass storage filesystem window

4. Open a browser (Firefox/Chrome) and access the url: http://192.168.7.2. The webpage hosted on the webserver of the board should indicate that the board is connected as shown below.

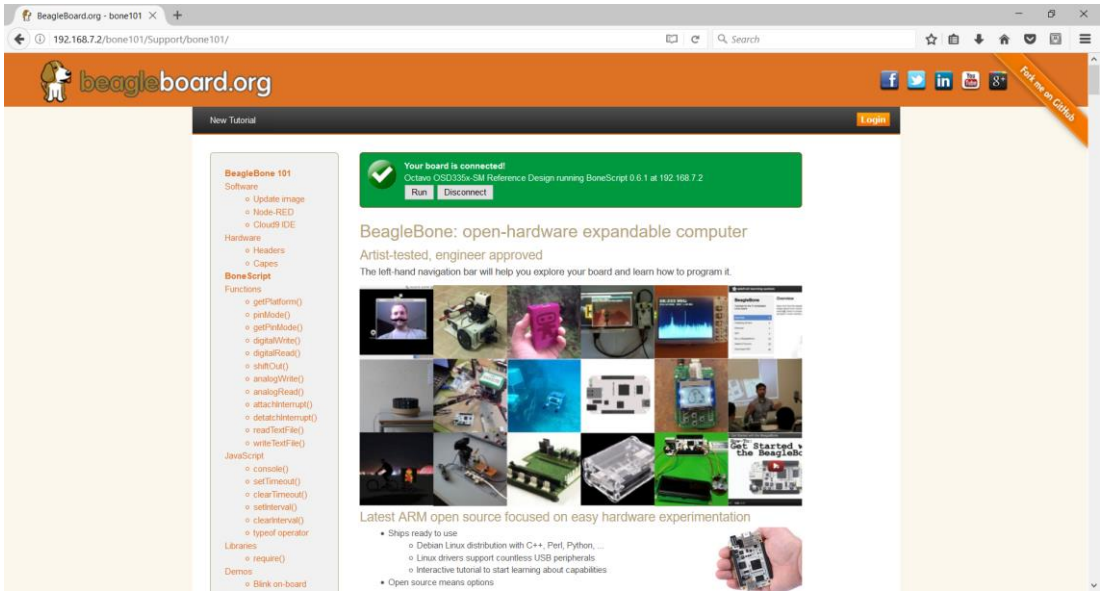

Figure 19: OSD3358-SM-RED home webpage indicating a connection to the board

5. Access the cloud9 IDE through the url: http://192.168.7.2:3000/. Cloud9 IDE is an opensource web based development environment supporting several languages. The environment should look like the following figure with workspace containing examples that can be executed. Cloud9 also loads a shell terminal that can be seen at the bottom of the figure.

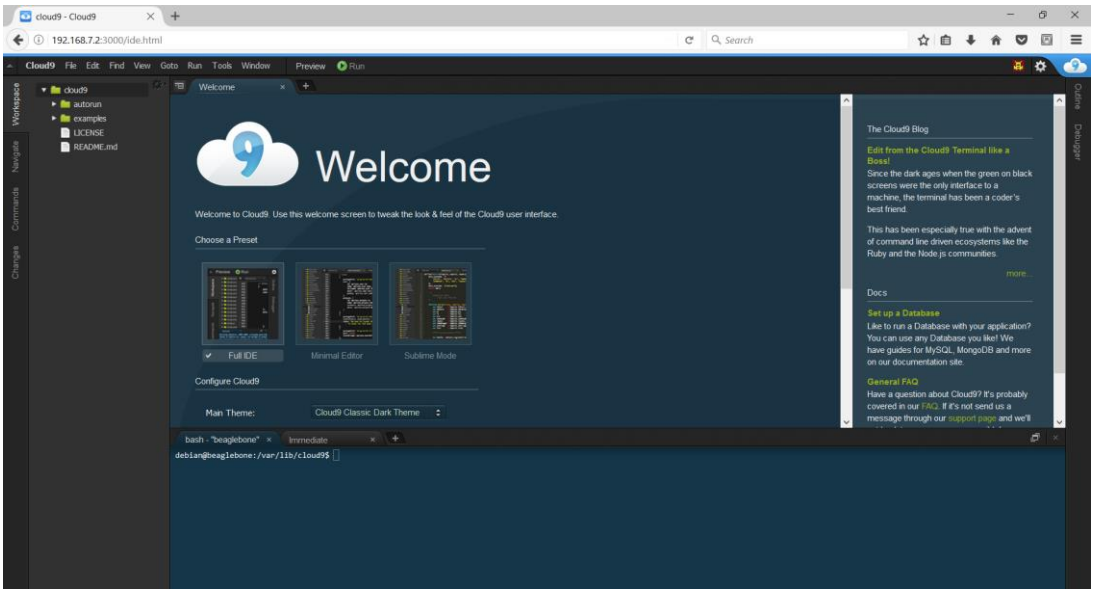

Figure 20: Cloud9 IDE environment

4.3 Boot messages through UART

The processor on the OSD3358-SM-RED platform sends boot messages through the UART0 port, J3. This is useful in embedded Linux development or if there are boot problems to verify proper initialization of all components on the board. After boot, the UART0 port will function as a Linux terminal. Accessing this requires a serial connection to a host device with a serial port. A USB-UART serial adapter can also be used for host devices with a USB port. A terminal program, such as Putty is required for communication.

The following steps describe the procedure to access the boot messages during boot and the terminal through UART0 port using a standard computer/laptop as a host device (Note: Given the computer/laptop has no native serial ports, a USB-UART serial adapter must be used):

1. Connect the UART side of the adapter to header J3 of the RED board. Make sure to check for proper orientation of the cable to connect Tx and Rx signals of the UART0 port appropriately to the adapter's signals.

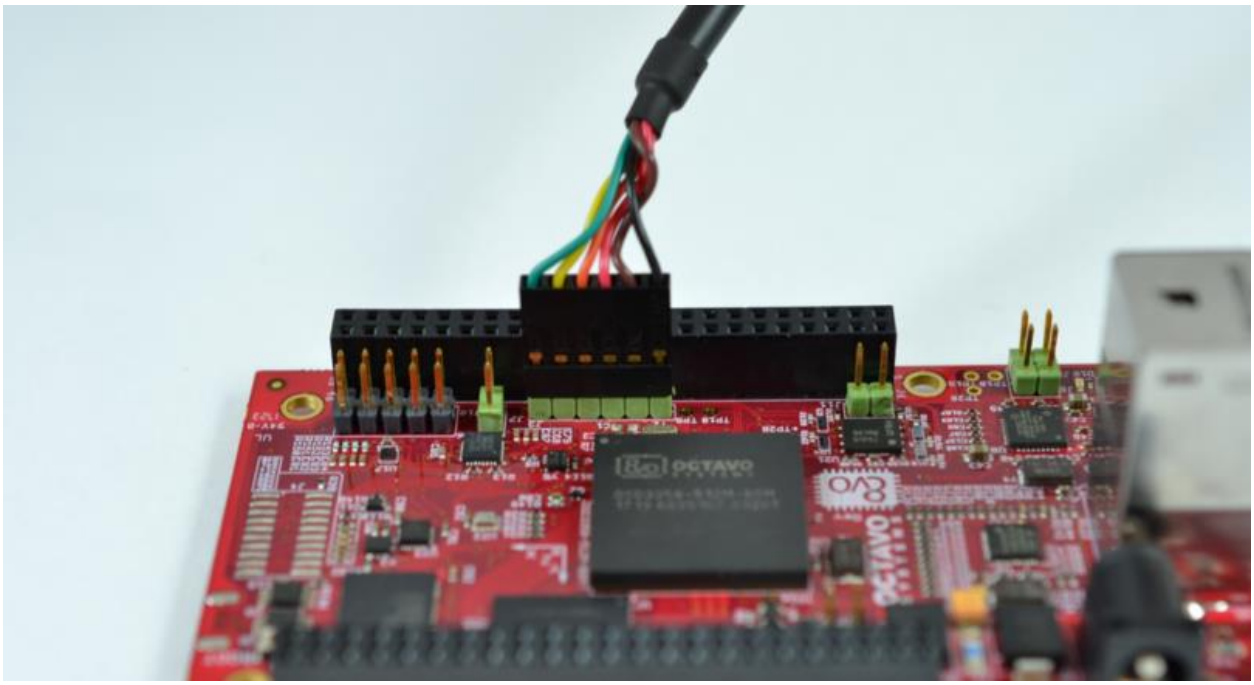

Figure 21: UART adapter cable plugged in to RED board

2. Connect the USB side of the adapter to the computer/laptop

Figure 22: UART-USB adapter plugged in to the computer

3. Check which COM port of the computer/laptop the adapter is connected to using the device manager.

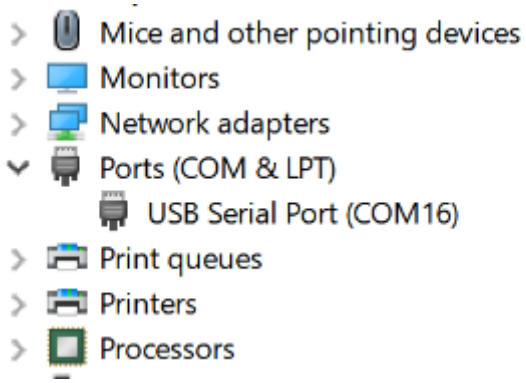

Figure 23: UART adapter detected as COM port 16

4. Open a serial console (opensource applications like Putty come with one). Select the serial communication option and specify the port and speed of the protocol as shown below and click on open.

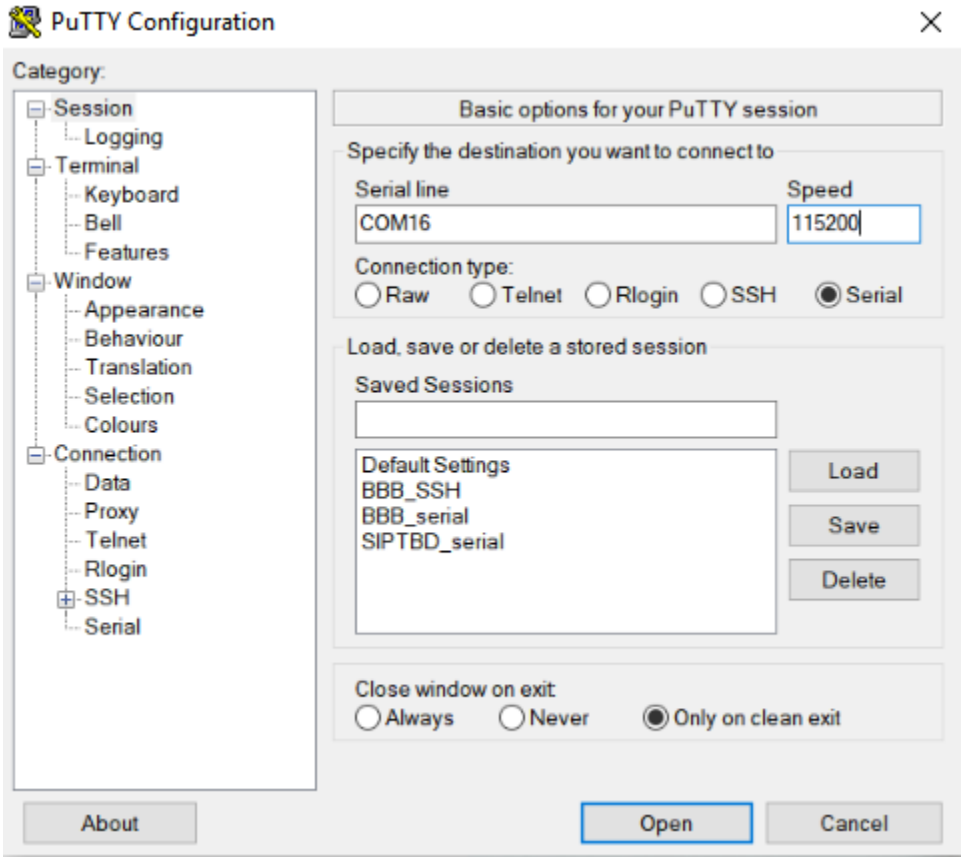

Figure 24: PUTTY configuration for serial communication

5. After a new serial console is open, connect the board to a power supply through one of the three power inputs of the board. The boot messages should begin to scroll on the console. After boot, the console will ask for secure shell login. You can login to the shell using the credentials – **debian:temppwd** .

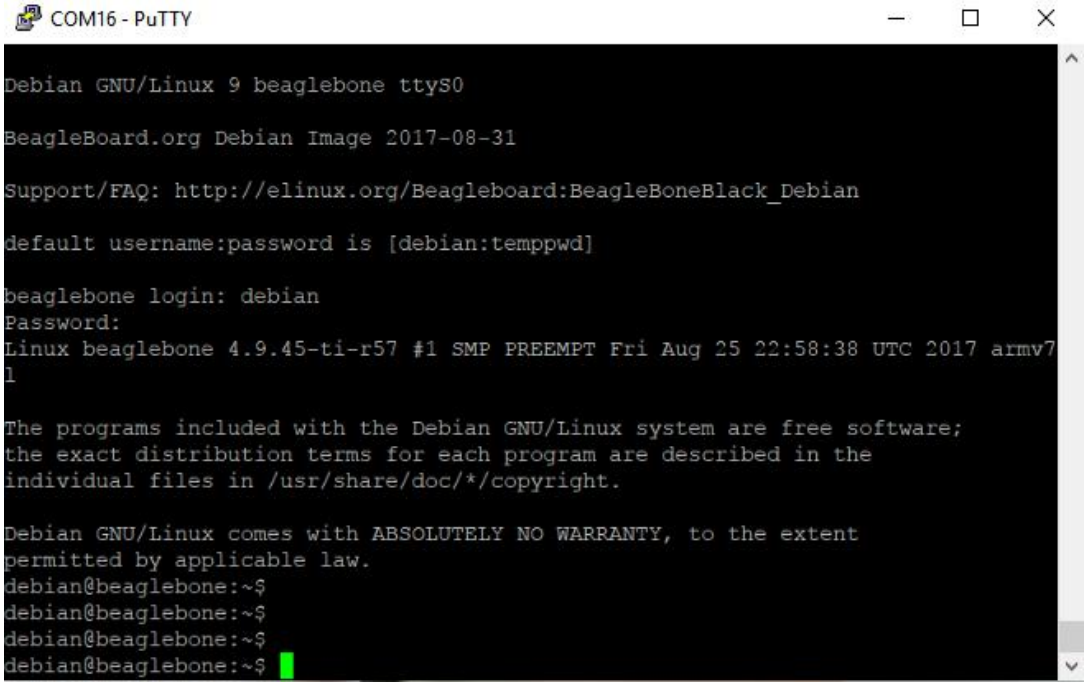

Figure 25: Debian Linux terminal after boot

4.4 User LEDs

While the LEDs D4 through D7 can be controlled as desired, they have the following standard function when the system is booted.

- D4: Activity indicator for MMC1 interface (eMMC)
- D5: Activity indicator for the kernel
- D6: Activity indicator for MMC0 interface (SD card)
- D7: Heartbeat indicator from Linux kernel

4.5 Powering down/power cycling

Once the system has completely booted, the board can be safely powered down in the following way:

1. Execute the shutdown command on Linux command line. Wait for LEDs D5 and D7 to turn off. Unplug the power source.

The board can be reset using one of the following methods:

1. Hold the PWR button S2 pushed for 8 seconds
2. Push the RST button S1

5 Getting started with the software

5.1 Onboard sensor interfacing

There are several sensors on OSD3358-SM-RED board. The board is equipped with a software library and examples to interface with the sensors. The library package 'Redperipherallib' is pre-installed on Linux and allows commands line access to these components on the board. The Roboticscape library (http://strawsondesign.com/#!manual-install) that comes pre-installed on the Beagleboard images was used as a base to build the Redperipherallib library. The library is located in **/home/debian/** folder of the Linux environment. The structure of the library is given below:

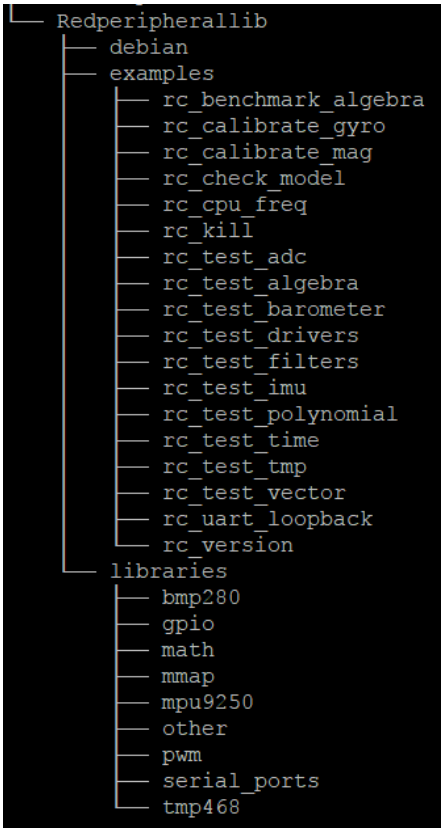

Figure 26: Structure of Redperipherallib

5.1.1 Accessing peripherals using shell terminal

The RED peripheral library facilitates command line access to board features. Some of the relevant commands, options and output formats to access on board sensor data are given below.

1. *rc_calibrate_gyro*
   Function: This command performs calibration of the gyro sensor in the IMU
   Options: None
   Result:

```
root@beaglebone:~$ rc_calibrate_gyro
Starting calibration routine

gyro calibration file written
run rc_test_imu to check performance
```

2. *rc_calibrate_mag*
   Function: This command calibrates the magnetometer in the IMU
   Options: None
   Result:

```
root@beaglebone:~$ rc_calibrate_mag
This will sample the magnetometer for the next 15 seconds
Rotate the cape around in the air through as many orientations
as possible to collect sufficient data for calibration
Press ENTER to continue or anything else to quit
spin spin spin!!!

keep spinning
you're doing great
keep spinning
you're doing great
keep spinning
you're doing great
Okay Stop!
Calculating calibration constants.....
Offsets X:  71.924 Y:  -1.718 Z: -24.156
Scales  X:   1.956 Y:   0.929 Z:   2.485
magnetometer calibration file written
run rc_test_imu to check performance
```

3. *rc_check_model*
   Function: This command returns the board name that it is currently running on
   Options: None
   Results:

```
root@beaglebone:~$ rc_check_model

Currently running on a:
OCT_OSD3358-SM-RED
```

4. *rc_test_barometer*
   Function: This command runs a test on the barometer and returns the temperature, pressure, altitude and filtered altitude sensor values
   Options: None
   Results:

```
root@beaglebone:~$ rc_test_barometer
  temp   | pressure   | altitude | filtered |
 35.08C | 101.57kpa |  -20.04m |  -19.88m |
```

5. *rc_test_imu*
   Function: This command tests and returns the IMU measurements
   Options:
   None : print sensor values with default settings
   -r: print raw values instead of radians
   -d: print gyro in deg/s instead of radians
   -h: print this help message
   Results:

```
root@beaglebone:~$ rc_test_imu
try 'test_imu -h' to see other options
   Accel XYZ(m/s^2)  |   Gyro XYZ (rad/s)  |  Mag Field XYZ(uT)  |
Temp (C)
  0.43   0.18   9.74 |   0.0   -0.0    0.0 |  -2.2  -78.4  -33.8 |
37.9
```

6. *rc_test_tmp*
   Function: This command tests the temperature sensor and returns the temperature value on channel 8
   Options: None
   Results:

```
root@beaglebone:~$ rc_test_tmp
temperature sensor read value in celsius: 28.750000
```

7. *rc_version*
   Function: This command returns the version of the peripheral library being run on the board
   Options: None
   Results:

```
debian@beaglebone:~$ rc_version
REDPeripherallib 0.3.4 built on Roboticscape 0.3.4
```